Supplementary information for cosi2 simulator

Ilya Shlyakhter, Pardis C. Sabeti, Stephen F. Schaffner

May 11, 2014

1 Approximating the coalescent: algorithm details

We say that a pair of nodes is coalesceable if the nodes' *extended convex hulls* overlap.

We maintain a dynamic collection of node hulls, as follows. We keep an augmented red-black tree of the hulls, indexed by hull beginnings. For each hull, we keep a count of hull intersections which begin at that hull; that is, the number of other hulls which start to the left and end to the right of the given hull's beginning. We augment the tree with subtree size information at each node, making it an order statistics tree.

Additionally, we keep an order statistics tree of hull ends. This allows us to quickly determine the number of hull beginnings or endings to the left or to the right of any given point.

Initially, all hulls have the form [0,1] (segment extending across entire simulated region). We ensure a total ordering of the hulls by breaking ties based on the order of addition to the data structure.

Adding a hull [B,E] involves two steps: we need to determine the number of new intersections starting at B (equal to the number of existing hulls that cross B); and for each existing hull beginning in the interval [B,E], we need to increment that hull's intersection count.

For the first step, we observe that it is easier to count hulls which do not intersect B: these are the hulls that either end before B, or start after B. We can determine these counts efficiently (in logarithmic time) using our order statistic trees of hull beginnings and ends. Subtracting the sum of these counts from the total number of hulls, we get the number of hulls that cross B; this will be the number of hull intersections that begin at B after the hull [B,E] is added.

1.1 Efficient range updates

For the second step, a naive implementation would loop over all existing hulls beginning in the interval [B,E], and increment their intersection counts. However, such a step would not be logarithmically bounded; in the worst case, it would need to individually visit a large number of hulls. Instead, we further augment

the order statistic tree of hulls with a δ field which is implicitly added to all intersection counts within the node's subtree. This lets us add an increment to the intersection counts of all nodes in a half-open interval, in logarithmic time. By adding +1 to the intersection counts of all nodes starting from B, and adding -1 to the intersection counts of all nodes to the right of E, we can increment all counts in the interval [B,E] with two logarithmic operations. Whenever we search for a node, as we traverse the path down from the root to the node, we keep track of the sum of delta fields, and adjust the intersection count stored in the node by this sum. When inserting a node, we likewise adjust the intersection count in the node by this sum, so that the implicitly represented intersection count of the new node equals its original count.

The δ values at the nodes must be maintained during rotations. This is done by ensuring, before each rotation, that the delta fields of nodes involved in a rotation are zero. A delta value at a node can be "pushed down" to its children by adding it to the delta values of the children, and zeroing the delta value at the node.

Removing a hull involves the reverse steps of adding a hull; their implementation is analogous.

All operations on the ancestral recombination graph – coalescence, recombination, gene conversion, migration – can be implemented in terms of hull addition and removal. Coalescence removes two existing hulls and adds one new one; recombination and gene conversion remove one existing hull and add two new ones; migration removes a hull from one hull pool and adds it to another. More efficient implementation of intersection count updates is possible in the case of coalescence, recombination and gene conversion, to perform a group of related hull additions and removals in a single step.

2 Correctness of the simulator, and accuracy of the approximation mode

We compared the distribution of a number of summary statistics computed for the output of *cosi2* (exact and approximate modes) and *msms*. The statistics included: π, the nucleotide diversity; ss, the number of segregating sites; D, Tajima's D; $\hat{\theta}_H$, Fay and Wu's H-statistic; bands of the allele frequency spectrum; and LD measures D' and r^2. The statistics were based on 10000 simulations of the following demographic model: effective population size, 30000; sample size, 80; simulated region length, 10MB; mutation rate, 1e-8; recombination rate, 1e-8. For each simulation, the value of the given statistic was computed; the empirical cumulative distribution functions of the 10000 values of the statistic were then compared. For linkage disequilibrium statistics, the statistic value for a simulation is taken to be the average of that statistic for SNP pairs separated by a specified number of SNPs.

Below are the summary statistics with the largest Kolmogorov-Smirnov deviations (D) between *cosi2* (exact) and *msms*:

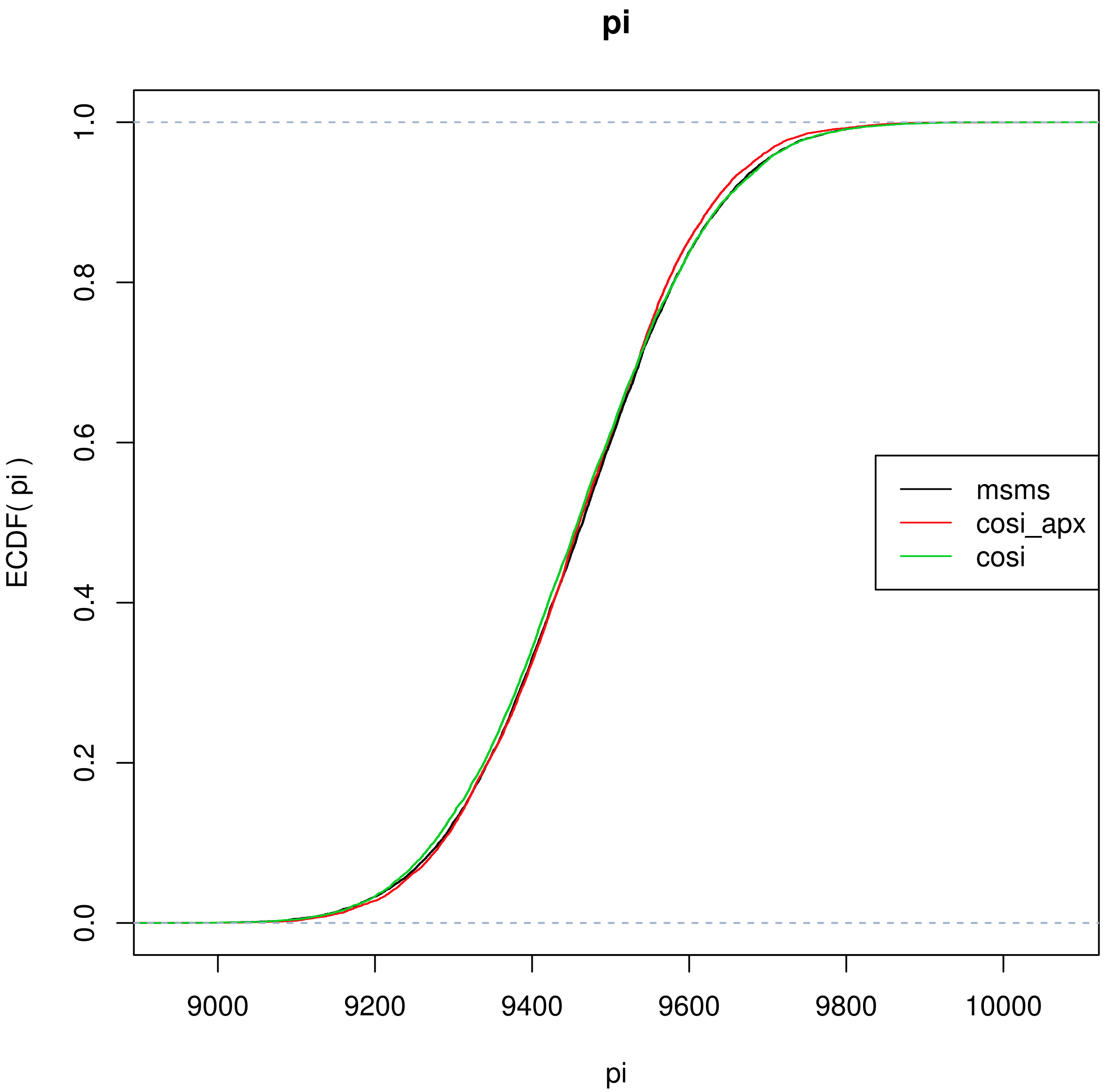
D	statistic
0.0216	ld_sep200_Dprime_mean
0.0198	afs_6_20_
0.0193	ld_sep100_Dprime_mean
0.0191	pi
0.0178	ld_sep10_Dprime_mean
0.0176	ld_sep5_Dprime_mean
0.0168	ld_sep5000_Dprime_mean
0.0164	ld_sep10000_r2_mean
0.0155	ld_sep100_r2_mean
0.0155	ss
0.0153	ld_sep10000_Dprime_mean
0.0148	ld_sep50_Dprime_mean
0.0144	ld_sep20000_Dprime_mean
0.0143	D
0.0142	ld_sep500_Dprime_mean
0.0142	theta
0.0137	ld_sep2000_Dprime_mean
0.0137	afs_71_80_
0.0136	ld_sep1000_Dprime_mean
0.0131	ld_sep500_r2_mean
0.0128	ld_sep20000_r2_mean
0.0118	afs_41_60_
0.0117	ld_sep1000_r2_mean
0.0111	ld_sep5000_r2_mean
0.0106	H
0.0104	afs_21_40_
0.0101	ld_sep200_r2_mean

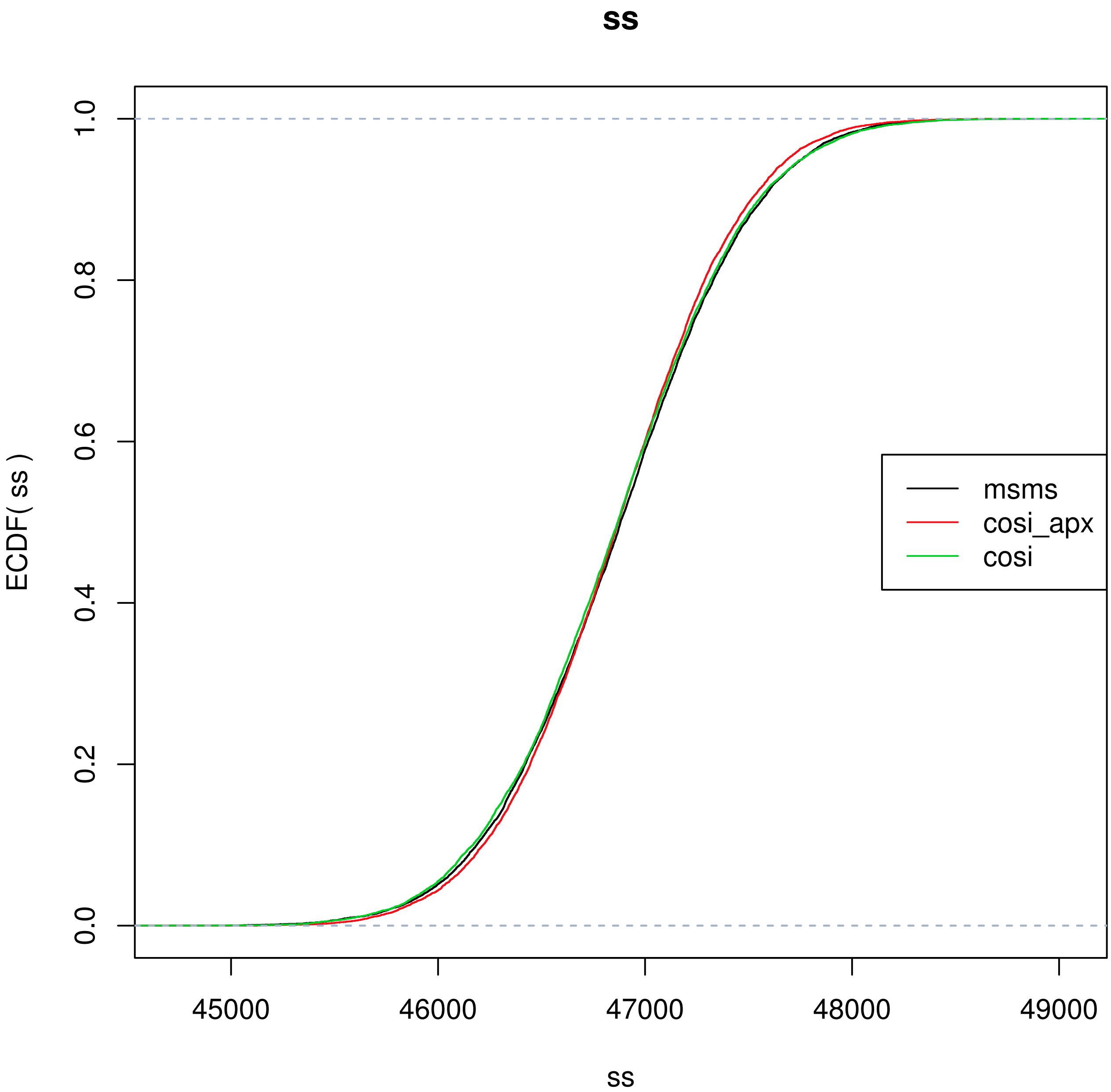
In the table, afs_M_N denotes the fraction of SNPs with derived allele count between M and N; ld_sepN_r2_mean denotes the mean r^2 for pairs of SNPs separated by N SNPs.

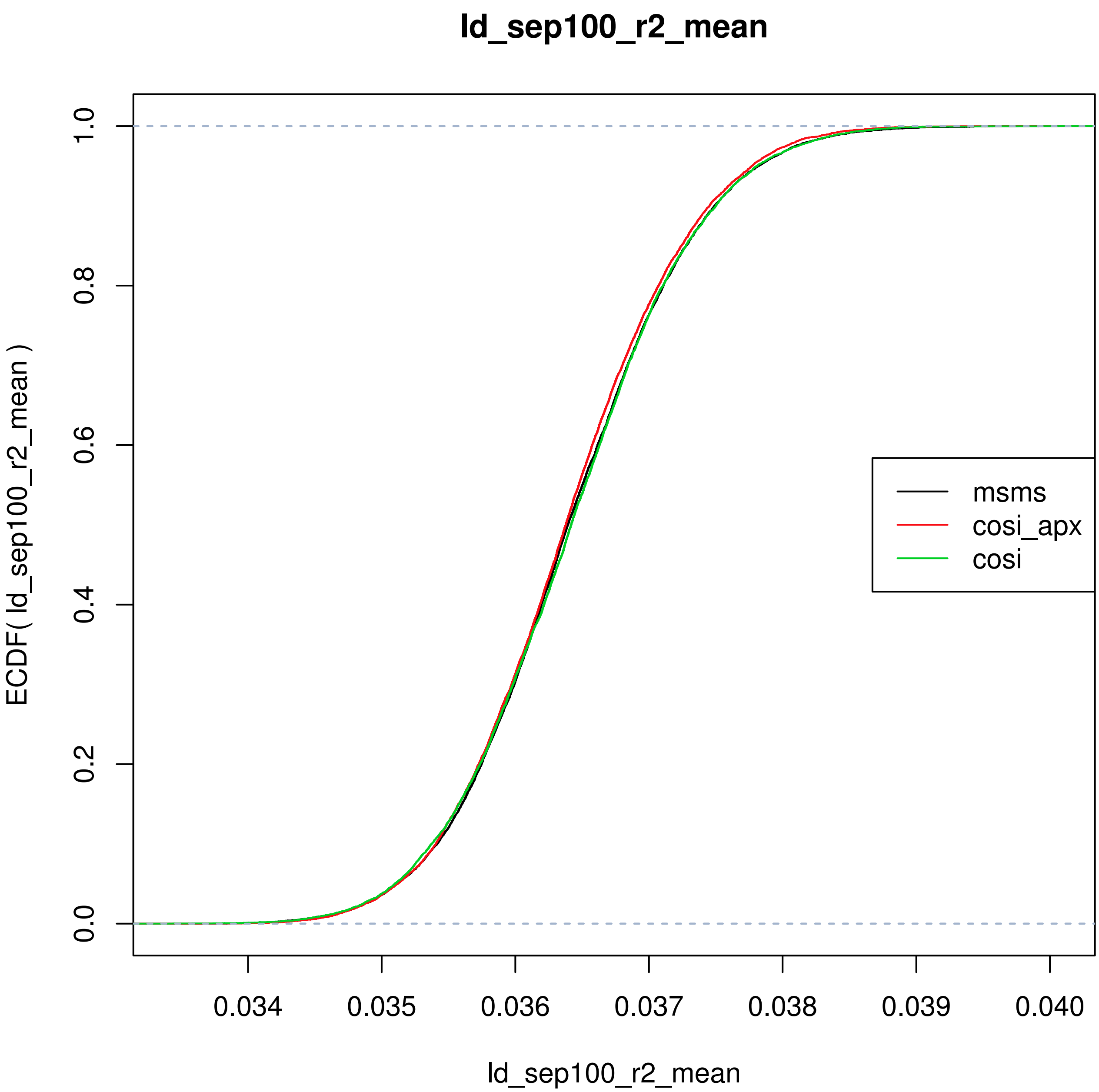
Following are the summary statistics with the largest Kolmogorov-Smirnov deviations (D) between *cosi2* (exact) and *cosi2* (approximate):

D	statistic
0.0292	ld_sep200_r2_mean
0.0276	ld_sep500_r2_mean
0.0268	ld_sep100_r2_mean
0.0233	ld_sep1000_r2_mean
0.0227	ld_sep5000_Dprime_mean
0.0222	ss
0.0202	pi
0.0187	ld_sep2000_Dprime_mean
0.0187	D
0.0181	ld_sep1000_Dprime_mean
0.0178	ld_sep50_Dprime_mean
0.0172	ld_sep2000_r2_mean
0.0172	afs_71_80_
0.017	ld_sep100_Dprime_mean
0.016	ld_sep500_Dprime_mean
0.016	ld_sep10000_r2_mean
0.0159	ld_sep20000_r2_mean
0.0159	ld_sep10_Dprime_mean
0.0156	ld_sep10_r2_mean
0.0155	ld_sep10000_Dprime_mean
0.0153	afs_41_60_
0.0151	ld_sep5_r2_mean
0.015	ld_sep5_Dprime_mean
0.0145	afs_21_40_
0.0144	afs_61_70_
0.0143	ld_sep20000_Dprime_mean
0.0142	theta
0.0126	ld_sep5000_r2_mean
0.0122	afs_1_
0.0121	ld_sep200_Dprime_mean
0.0118	ld_sep50_r2_mean
0.0118	afs_6_20_
0.0106	H
0.0081	afs_2_5_

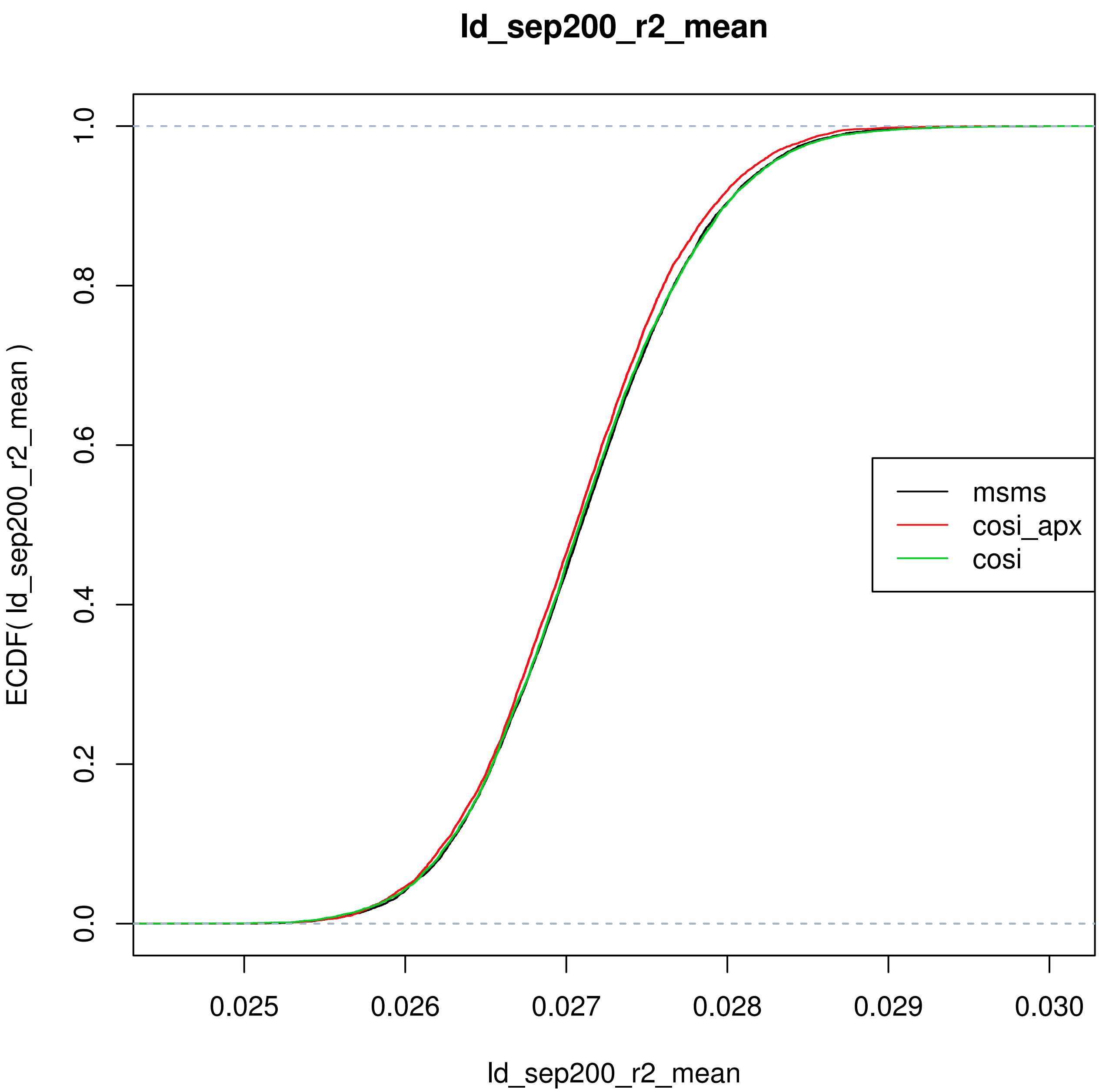
Following are the empirical cumulative distribution function plots for selected statistics, including ones showing the largest deviations. The complete set of plots, as well as additional comparisons, can be downloaded from the *cosi2* website at `http://broadinstitute.org/~ilya/cosi2`.

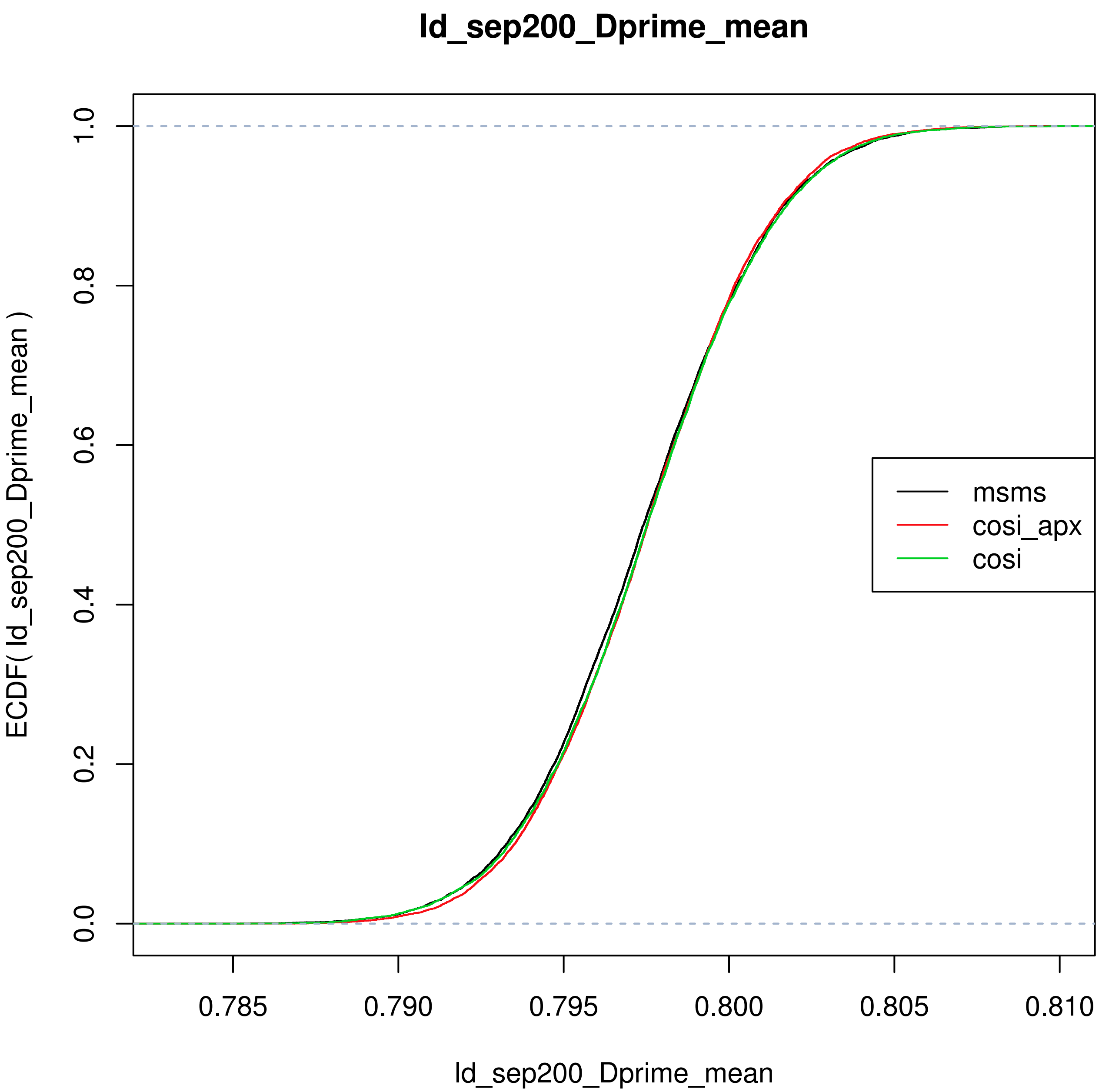

pi
1.0
0.8
0.6
0.4
0.2
0.0
ECDF(pi)
9000
9200
9400
9600
9800
10000
pi
msms
cosi_apx
cosi

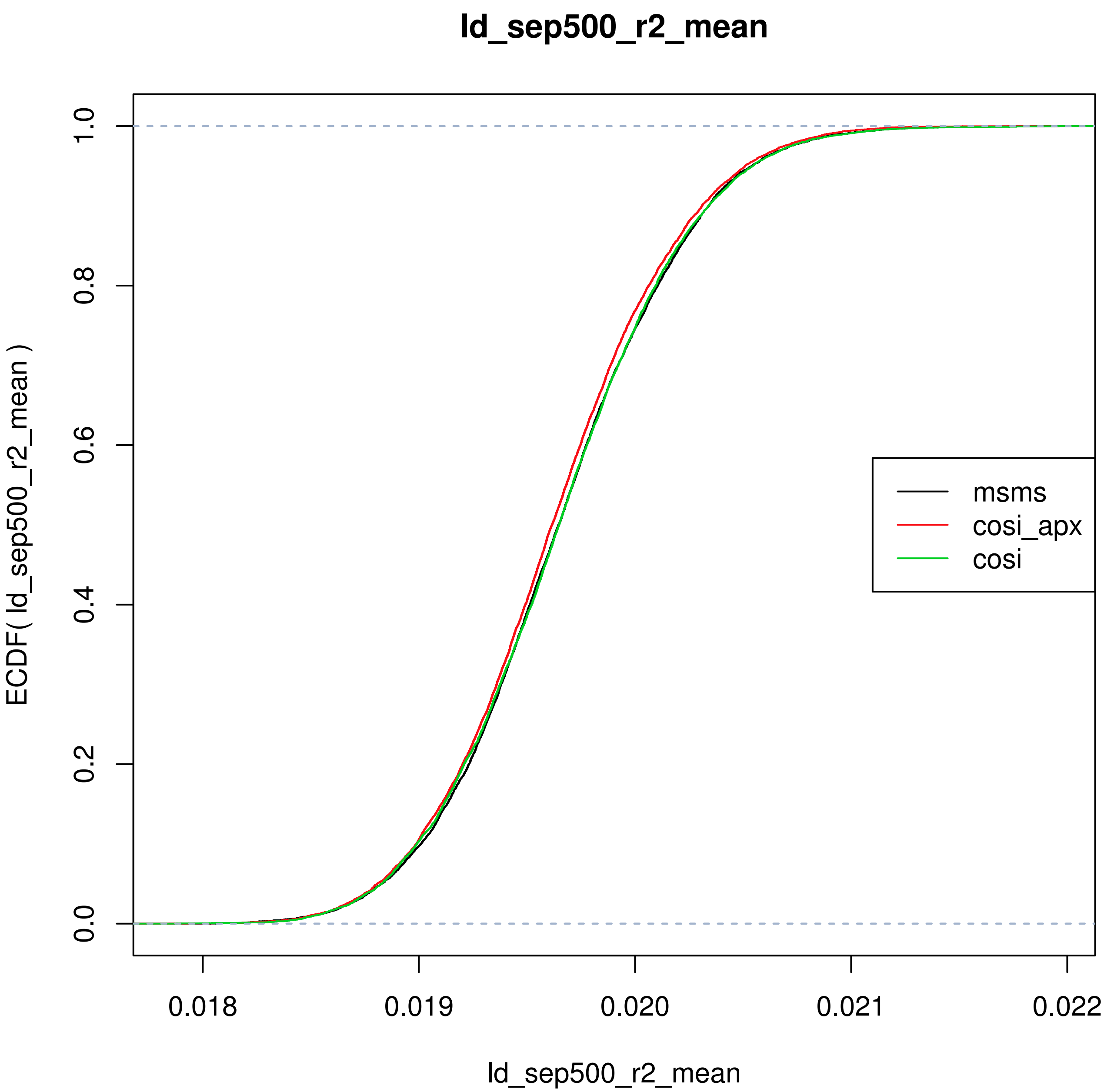

ss
ECDF(ss)
1.0
0.8
0.6
0.4
0.2
0.0
45000
46000
47000
48000
49000
ss
msms
cosi_apx
cosi

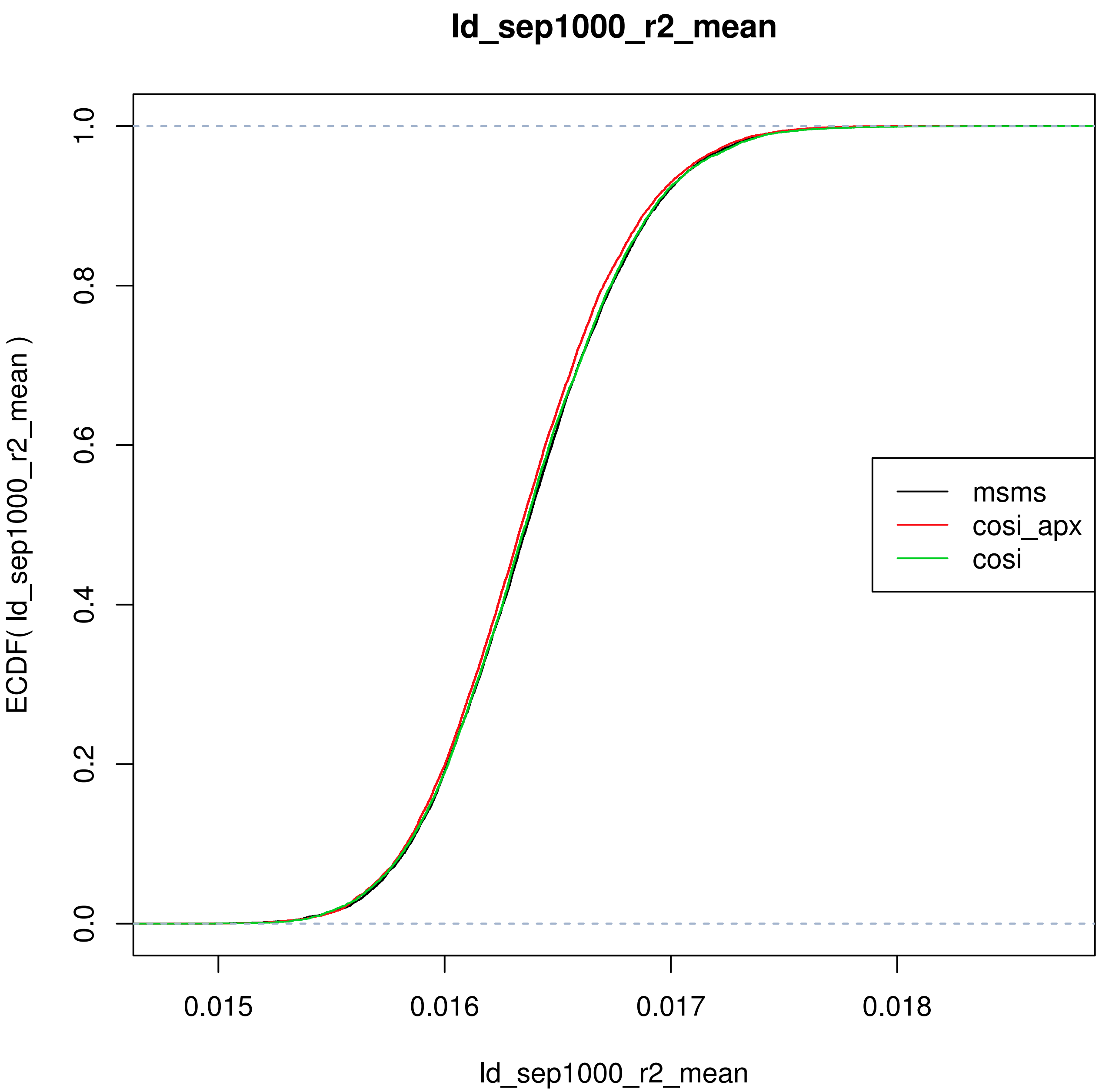


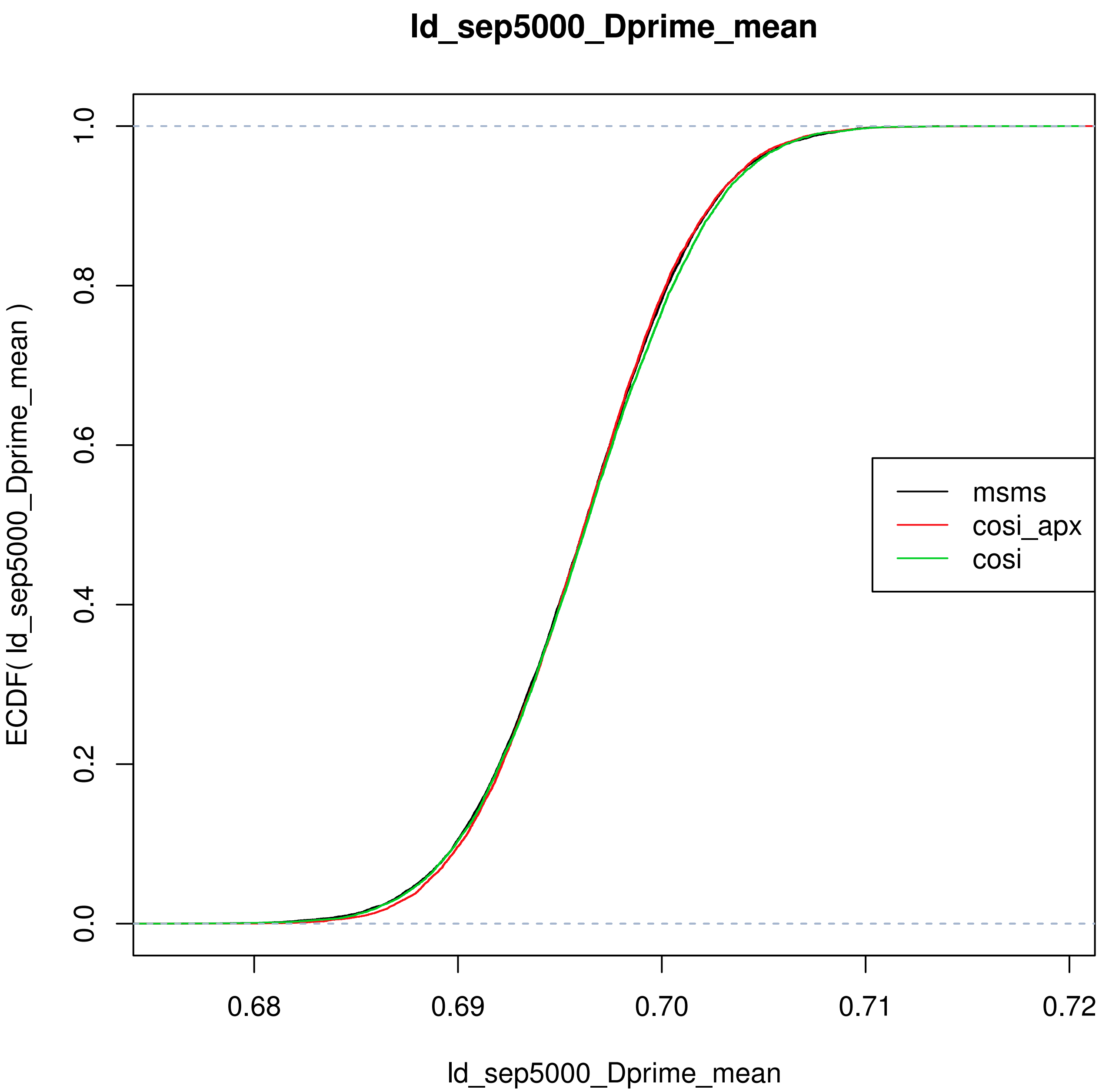
ld_sep100_r2_mean
ECDF(ld_sep100_r2_mean)
0.0
0.2
0.4
0.6
0.8
1.0
0.034
0.035
0.036
0.037
0.038
0.039
0.040
ld_sep100_r2_mean
msms
cosi_apx
cosi


ld_sep200_r2_mean
1.0
0.8
0.6
0.4
0.2
0.0
ECDF(ld_sep200_r2_mean)
msms
cosi_apx
cosi
0.025
0.026
0.027
0.028
0.029
0.030
ld_sep200_r2_mean


ld_sep200_Dprime_mean
ECDF(ld_sep200_Dprime_mean)
ld_sep200_Dprime_mean
0.0
0.2
0.4
0.6
0.8
1.0
0.785
0.790
0.795
0.800
0.805
0.810
msms
cosi_apx
cosi


ld_sep500_r2_mean
ECDF(ld_sep500_r2_mean)
0.0
0.2
0.4
0.6
0.8
1.0
0.018
0.019
0.020
0.021
0.022
ld_sep500_r2_mean
msms
cosi_apx
cosi


ld_sep1000_r2_mean
ECDF(ld_sep1000_r2_mean)
0.0
0.2
0.4
0.6
0.8
1.0
0.015
0.016
0.017
0.018
ld_sep1000_r2_mean
msms
cosi_apx
cosi


ld_sep5000_Dprime_mean
ECDF(ld_sep5000_Dprime_mean)
1.0
0.8
0.6
0.4
0.2
0.0
0.68
0.69
0.70
0.71
0.72
ld_sep5000_Dprime_mean
msms
cosi_apx
cosi